REMARKS

By this amendment, claims 1-35 and 37 are pending, in which claim 37 is newly added, claim 36 is canceled without prejudice or disclaimer and claims 1, 5-6, 16, 20-21, 31-32, and 35 are currently amended. No new matter is introduced.

The Office Action mailed May 7, 2003 rejected claims 1-5, 16-20, and 31-36 as obvious under 35 U.S.C. § 103 based on *Li et al.* (US 5,774,588) in view of the newly-cited *Rangarajan et al.* (US 5,706,365). Claims 15 and 30 have been allowed and claims 6-14 and 21-29 have been objected to as being allowable but dependent on a rejected base claim.

In response to the objection, claims 6 and 21 have been rewritten into independent form. Accordingly, claims 6-14 and 21-29 are now in condition for allowance.

The rejection of claim 35, which has been amended to recite that the lexical containers are "each configured to contain a respective maximum number of entries based on a function that includes a term that is inversely proportional to logarithm of a key length associated with the lexical containers" (see Specification, p. 13, formula (2), line 15, for adequate descriptive support), is respectfully traversed. No disclosure of logarithms can be found in the applied references, and the Office Action did not point where a logarithm can be found in the rejection of previously presented claim 36, which expressly recited a "logarithm."

The rejection of claims 1-5, 16-20, and 31-36 is respectfully traversed because neither *Li et al.* nor *Rangajan et al.*, individually or in combination, teach the elements of the claims. For example, independent claims 1 and 31, as amended, recites (with added emphasis) as follows, with independent computer-readable medium claims 16 and 32 amended to mirror the steps of method claims 1 and 31, respectively:

1. A method of searching for a string in a lexical cache, comprising the computer-implemented steps of:

generating a key based on the string;

selecting a lexical container from among a plurality of lexical containers based on a length of the key, said lexical containers associated with respective key lengths and configured to hold respective maximum numbers of entries based on the respective key lengths; and

searching the selected lexical container for an entry associated with the string based on the key,

wherein at least one of the lexical containers is configured to hold a different maximum number of entries than at least another one of the lexical containers.

31. A method of storing a string in a lexical cache, comprising the computer-implemented steps of:

generating a key based on the string;

selecting a lexical container from among a plurality of lexical containers based on a length of the key, said lexical containers are associated with respective key lengths and configured to hold respective maximum numbers of entries based on the respective key lengths; and

storing the string in an entry in the selected lexical container based on the key,

wherein at least one of the lexical containers is configured to hold a different maximum number of entries than at least another one of the lexical containers.

This limitation is not found in either *Li et al.* or *Rangarajan et al.* Rather, no data structure in *Li et al.* or *Rangarajan et al.* that satisfy the claim limitations that a lexical container is selected "based on a length of a key." In particular, *Li et al.* is directed to a method for comparing strings with entries of a lexicon using a fixed-length key. At step **120** of FIG. 1B, an incoming, unverified string **20** is processed to produce a signature vector **25** having a fixed length of 85 bits (col. 6:59-67). Then seven 12-bit partitions **30-36** of the signature vector **25** are hashed (FIG. 2, step **240**; col. 7:63–8:14) are hashed to generate a bucket address, with which entries of the *Li et al.* lexicon are stored and retrieved. Associated with each bucket address is a linked list of buckets, in which each bucket holds a fixed number of entries, and the bucket list expands "to accommodate multiple signature vectors indexed to that address." (col. 7:56-58).

By contrast, *Li et al.* does not disclose selecting a lexical container or even a hash table from among a plurality of such based on "a length of the key" as recited in the claims, because the key length is fixed to a constant length. The signature vector **25** key is fixed at 85 bits, and the partitions **30-36** of the signature vector **25** are a fixed 12 bits in length. Thus, there is no need nor motivation in *Li et al.* to use that key's constant length to identify a particular lexical container or hash table.

Moreover, *Li et al.*'s linked list of buckets at each bucket address, upon which the Office Actions reads the recited "lexical container," fails to satisfy the limitations recited for the lexical container. For example, *Li et al.* fails to disclose that the linked list of buckets at each bucket address is "associated with respective key lengths" or "configured to hold respective maximum numbers of entries based on the respective key lengths." In fact, the Office Action acknowledges that "Li does not explicitly indicate a length of the key, and wherein at least one of the lexical containers is configured to hold a different maximum number of entries than at least another one of the lexical containers." (p. 4, emphasis original)

The newly cited *Rangarajan et al.* is similarly deficient. *Rangarajan et al.* describes a system and method for "indexing and retrieval of stored documents using a decomposition of words in the documents in n-grams, or linear word subunits." (Abstract). With specific reference to FIG. 10 for the indexing aspect of *Rangarajan et al.*, each word is expanded into *k* different n-grams (step **1011**), and an n-gram number is determined for each of the *k* n-grams (step **1013**). The group consisting of *k* and the *k* n-gram numbers is called a "word key" (col. 8:60–9:8). The n-gram number is used to index into a index page map within a bank index **223** (step **1023**) in order to set a corresponding bit in the index page map (step **1025**). After this loop is complete, the *Rangarajan et al.* system generates and writes a page key **509** (step **1031** or **1033**). The page key **509** denotes the "set of word keys for the all [*sic*] words on a page" (col. 9:7-8).

In retrieval, *Rangarajan et al.* discloses that a query word is decomposed into n-grams (FIG. 12, step **1205**), and, if an n-gram is present in a bank **217** (steps **1207**), the index page map for each page stored in the bank is checked for the n-gram (steps **1209** and **1211**). If the bit is set, a counter is set (step **1213**), and if the counter is sufficient to consider the page a hit, the page is flagged as a hit (step **1217**). As shown in FIG. 14, a page flagged as a hit is further searched, by looping over each n-gram in the Query Word (step **1403**), visiting each word key in the page (step **1405**) to determine if a sufficient number of n-grams in the Query Word and the word key match (steps **1407-1413**).

By contrast, *Rangarajan et al.* fails to disclose "selecting a lexical container from among a plurality of lexical containers **based on a length of the key**." Rather, in *Rangarajan et al.* a page is flagged as a hit when n-grams of the Query Word are found in the page key **509**. Furthermore, *Rangarajan et al.* fails to suggest "said lexical containers associated with respective key lengths" because each page key **509** is associated instead with each page of a document that is indexed in the *Rangarajan et al.* system and has nothing to do with the length or number of n-grams in the Query Word. The page key size **513**, cited in the Office Action, does not support the rejection, because the page key size **513** is not "a length of a key" (rather it is the size of all the words keys in the page) nor the criterion used to select a page key **509** in the *Rangarajan et al.* system, which is matching n-grams.

The dependent claims are allowable for at least the same reasons as their independent and are separately patentable on their own merits. For example, claim 37 recites "searching only the selected lexical container"; however, all pages that are flagged as hit in *Rangarajan et al.* are searched. As another example, dependent claims 33-34 recite that "the first key length is less than the second key length; and the first lexical container is configured to hold more entries than the second lexical container." Although the Office Action asserts that the second portion of the

Li et al. lexicon hold some of the entries of the first portion of the lexicon (pp. 5-6 citing col. 4:29-33 of *Li et al.*), there is nothing in *Li et al.* or the Office Action that the first portion of the lexicon is associated with a first key length that is less than the second key length for the second portion of the lexicon. In fact, the Office Action correctly acknowledges that "Li does not explicitly indicate a length of the key."

Therefore, the present application, as amended, overcomes the rejections of record and is in condition for allowance. Favorable consideration is respectfully requested. If any unresolved issues remain, it is respectfully requested that the Examiner telephone the undersigned attorney at 703-425-8516 so that such issues may be resolved as expeditiously as possible.

Respectfully Submitted,

DITTHAVONG & CARLSON, P.C.

7/29/2003
Date

Stephen C. Carlson
Attorney/Agent for Applicant(s)
Reg. No. 39929

10507 Braddock Rd
Suite A
Fairfax, VA 22032
Tel. 703-425-8516
Fax. 703-425-8518